



Universidad
de Concepción

Tecnologías de Información

Departamento de Ingeniería
Informática y Ciencias de la
Computación





Diseño de Base de Datos



Fases del Diseño de Base de Datos

Modelamiento de Datos

Fases del Diseño de Base de Datos

Contextualización de Los Requisitos

Esquema Conceptual a través de un Modelo de Datos

Requisitos Funcionales

Diseño Final

- ▶ Diseño Lógico
- ▶ Diseño Físico

Esquema Conceptual a través de un Modelo de Datos

Se elige un tipo de Modelo de Datos, para representar las necesidades e información a almacenar en la Base de Datos del sistema que se está implementando. Se hace necesario identificar:

- ▶ Entidades
- ▶ Relaciones
- ▶ Atributos

Evitar Redundancia e Incompletitud

Modelo Entidad Relación

Muchas similitudes con otros lenguajes de modelado como UML.
Conceptos claves son:

- ▶ *Entidades / Tipo de entidad, Atributos, Relaciones / Tipo de relaciones, Restricciones.*

Más cerca de la forma en que la gente piensa.

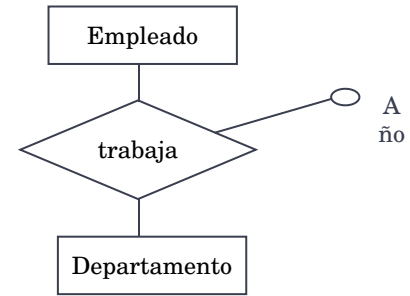
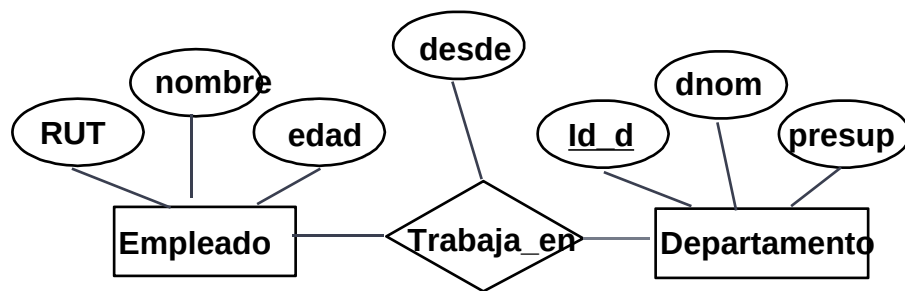


Diagrama de Datos E-R (Modelo Entidad Relación, MER)

Un diagrama de entidad-relación es un gráfico con nodos que representan tipos de entidades, atributos, y tipos de relaciones.



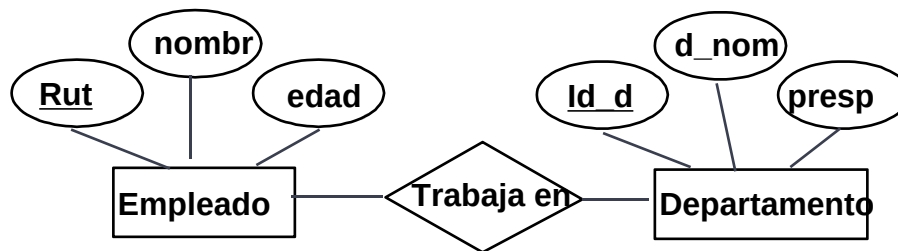
Entidades y Tipos de Entidad

Entidad: objeto del mundo real distinguible de otros objetos

Una entidad está asociada con los atributos que describen sus propiedades.

Los valores de atributo son atómicos, pero también existe la posibilidad de indicar atributos compuestos

Tipo de entidad: una colección de entidades similares.



Relaciones y Tipo de Relación

Relación: asociación entre dos o más entidades.

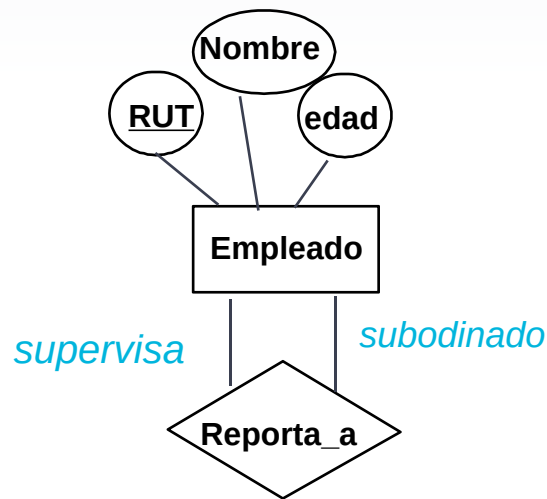
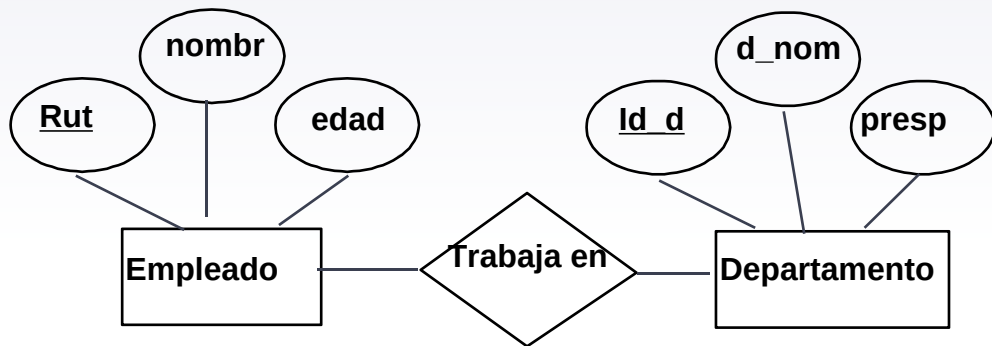
Tipo de relaciones: Colección de relaciones similares entre dos o más tipos de entidades.

Una relación n-aria R se relaciona con n conjuntos de entidad $E_1 \dots E_n$.

Los tipo de relación binaria son los más comunes.

El mismo tipo de entidad puede participar en diferentes tipo de relaciones o en diferentes "**roles**" en el mismo conjunto.

Relaciones y Tipo de Relación



Ejemplos

Entidad

- ▶ Objeto que se distingue de otros objetos su domicilio,
- ▶ Curso 503 354

Tipo de entidades

- ▶ Todas las direcciones domésticas
- Colección de cursos

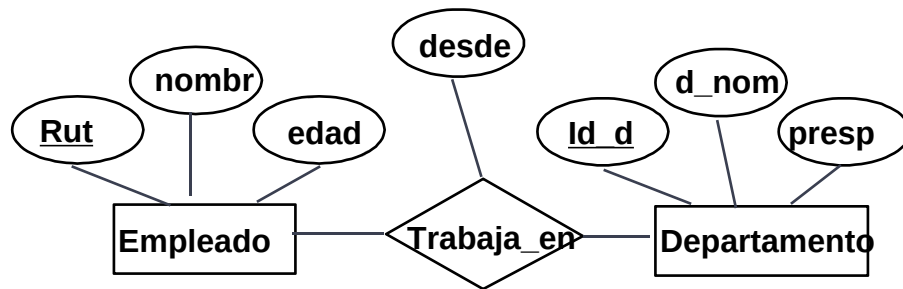
Relación

- ▶ Joe vive en 45 Main St.
Mary vive en 89 Wood Ave.

Relaciones y Tipo de Relación

Los tipos de relaciones también pueden tener **atributos**.

Útil para propiedades que no se pueden asociar razonablemente a uno de los conjuntos de entidades participantes.



Instancias de un diagrama

ER

Un Tipo de Entidad contiene un conjunto de entidades. Cada entidad tiene un valor para cada uno de sus atributos.

No hay instancias duplicadas

Empleados

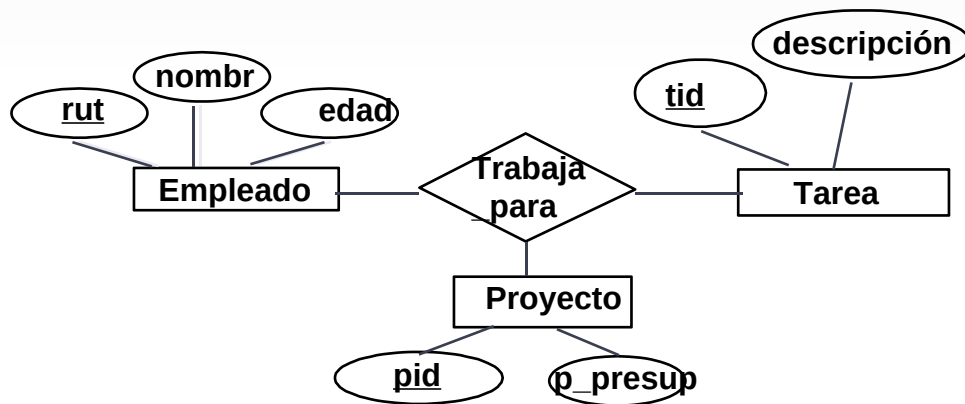
rut	Nombr	edad
12345678	"Pablo Muñoz"	30
14789632	"Camila Pérez"	25
.

Atributo

- Cada una de las propiedades o características que tiene un tipo de entidad o un tipo de relación corresponde a un atributo.
- Una cierta característica o propiedad de un objeto y toman valores que pertenecen a un determinado dominio
- Son elementos de un dominio. Aporta mediante su rótulo o nombre, la semántica de los valores del dominio al que ésta asociado
- Conjunto de valores de un mismo tipo. Se define un conjunto, ya sea de extensión o comprensión.

Relaciones y Tipo de Relación

Los tipo de relaciones *múltiples* ($n > 2$) se utilizan siempre que las relaciones binarias no puedan capturar la semántica de la aplicación.



Infrecuentes

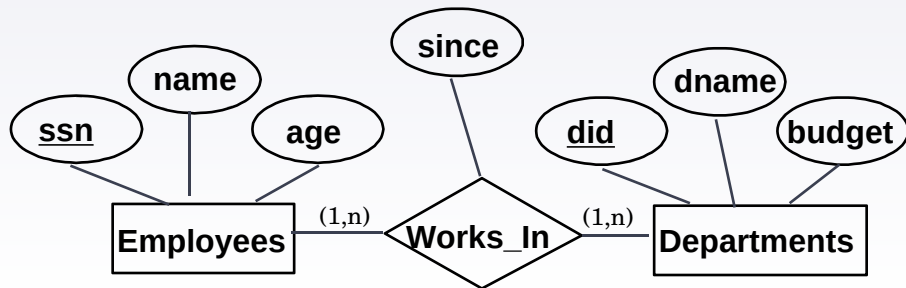
► Modelamiento de las Restricciones

En el modelo de datos se pueden definir ciertas restricciones que tiene el sistema que estamos diseñando. Típicamente, estas son:

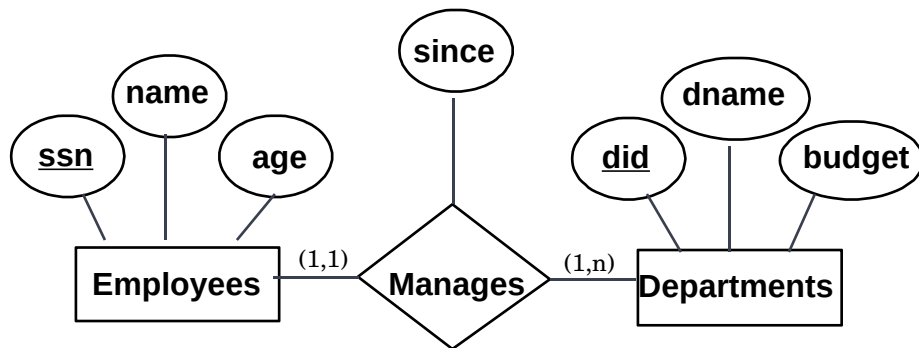
- ▶ Correspondencia de Cardinalidades
- ▶ Identificación de Claves (Candidatas, Primarias)
- ▶ Participación (de entidades en una Relación)

Cardinalidad de Relaciones

Un empleado puede trabajar en *muchos* departamentos; un departamento puede tener *muchos* empleados.



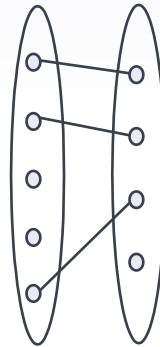
Cada departamento *tiene como máximo* un empleado, que puede administrar *varios* (*muchos*) departamentos.



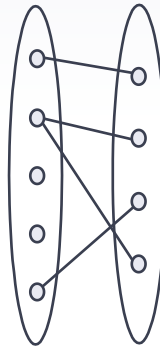
Cardinalidad de Relaciones

Los diferentes tipos de relaciones (binarias) desde un punto de vista de multiplicidad:

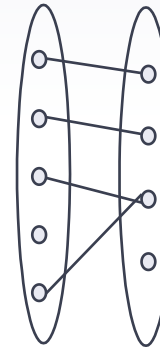
- ▶ Uno a uno
- ▶ Uno a muchos
- ▶ Muchos a uno
- ▶ Muchos a muchos



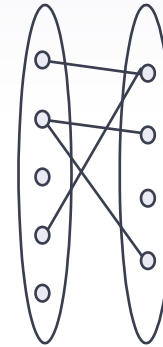
uno a uno



uno a muchos



varios a uno

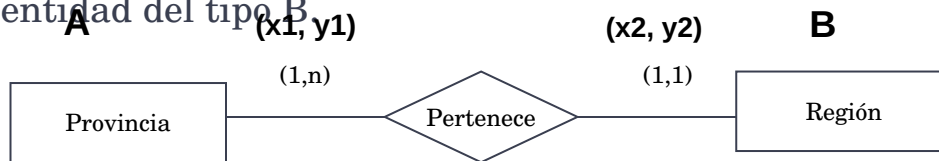


muchos a muchos

Cardinalidad de Relaciones (Nomenclatura Crow's Foot and UML)

La cardinalidad expresa la cantidad mínima y máxima de la ocurrencia de una entidad en la entidad relacionada.

En un Diagrama Entidad Relación denotaremos la cardinalidad con el par $(x1,y1)$ al costado de la entidad A, donde “x1” representa la cantidad mínima de esta entidad con la cual se asocia cada entidad de tipo B e “y1” representa la cantidad máxima de entidades tipo A con que se puede asociar cada entidad del tipo B.



En este caso el par $(x1=1,y1=n)$ adosado a la entidad de tipo Provincia (A), indica que una entidad de tipo Región (B) cómo mínimo “posee” (inversa de pertenencia) 1 Provincia y cómo máximo “n” Provincias. Análogamente el par $(x2=1,y2=1)$ adosado al tipo de entidad Región, nos indica que toda entidad de tipo Provincia “Pertenece” como mínimo a una $(x2=1)$ entidad tipo Región, pero también como máximo a una sola Región $(y2=1)$.

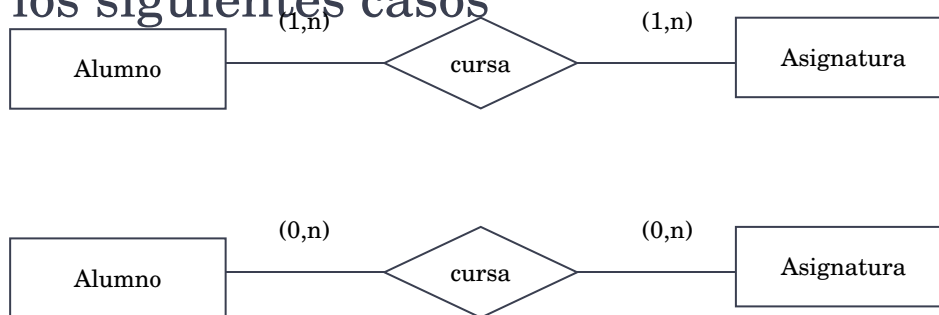
Cardinalidad

Relaciones (“1 a n” o “1:n”) . Una ocurrencia de la entidad de la izquierda está asociada con muchas (n) de la otra. En este caso generaliza cualquiera de las siguientes situaciones:



► Cardinalidad

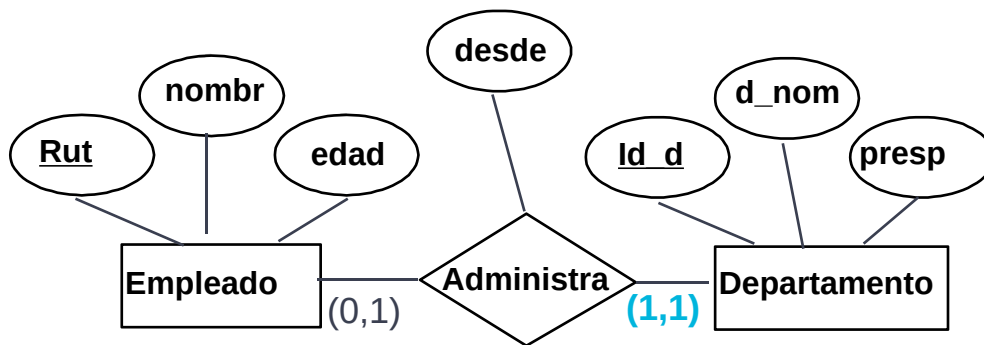
Relaciones (“n a n “ o “n:n”) Cada ocurrencia, en cualquiera de las dos entidades de la relación, puede estar asociada con muchas (n) de la otra y viceversa. Esto sería una generalidad de cualquiera de los siguientes casos



Lo anterior, es porque cuando se generaliza, nos centramos en los máximos u no en los mínimos. Análogo se aplica, cuando se habla de “varios a uno” y “uno a varios”.

Restricción de Clave

Una **restricción de clave** en un conjunto de relaciones especifica que el conjunto de entidades marcadas participa **como máximo** en una relación de este conjunto de relaciones.



Restricción de Participación

Una **restricción de participación** en un conjunto de relaciones, especifica que una o más entidades, debe participar en una determinada relación, lo que se traduce en que la cardinalidad mínima de esa entidad sobre la entidad relacionada es 1.

En este caso una Clase no necesariamente debe ser generada por un Curso.

CLASS is optional to COURSE



En este caso toda Clase es generada por un Curso y un Curso debe generar, al menos una Clase.

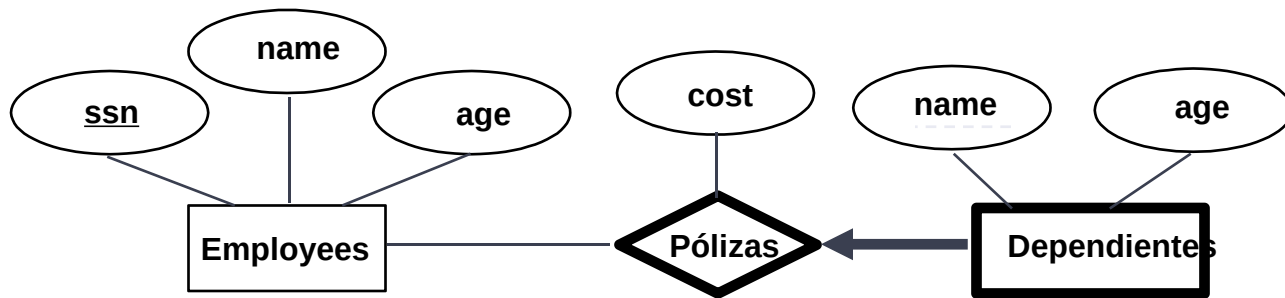
COURSE and CLASS in a mandatory relationship



Entidades Débiles

Una **entidad débil** existe solo en el contexto de otra entidad (**propietaria**).

La entidad débil solo se puede identificar de forma única si se considera la **clave principal del propietario** y su **propia clave parcial**.



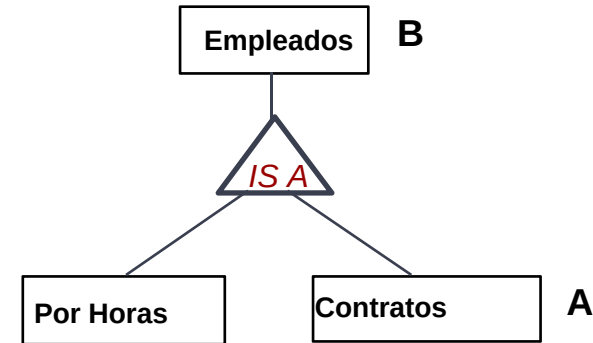
Subclases

A veces, un tipo de entidad contiene algunas entidades que comparten muchas, pero no todas las propiedades con la entidad establecen jerarquías.

A **IS A (Es una)** B: cada entidad A también se considera una entidad B. A especializa B, B generaliza a A.

A se llama subclase,
B se llama superclase.

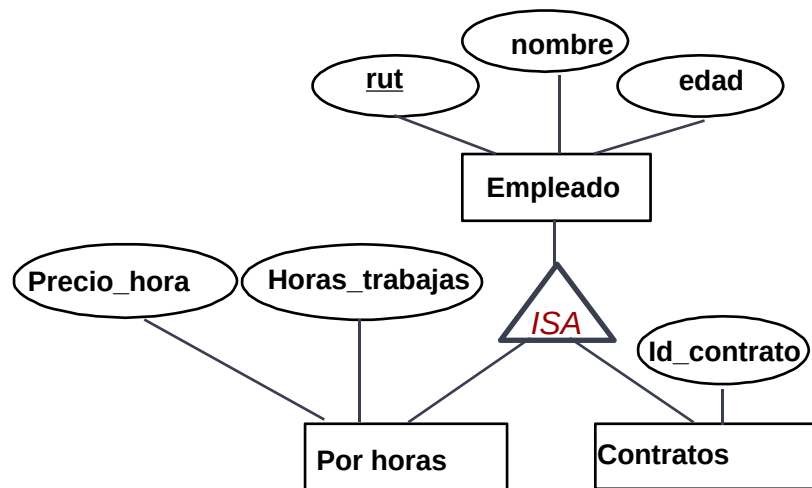
Una subclase hereda los atributos de una superclase y puede definir atributos adicionales.



Subclases

Por horas y Contratos heredan el rut (Clave!), el nombre y los atributos de edad de Empleado.

Definen atributos adicionales precio_hora, horas trabajadas y id_contrato respectivamente.



► Subclases - Cobertura

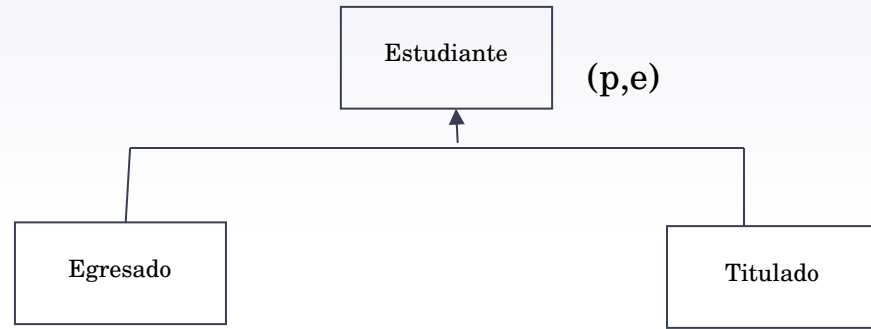
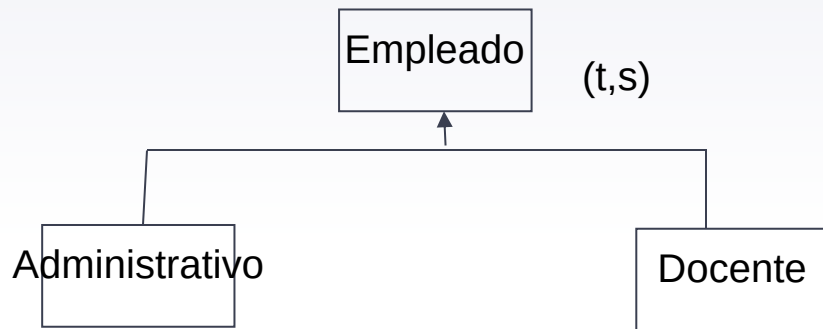
Cobertura total o parcial

Todos los elementos del tipo de entidad genérico deben pertenecer a alguno de sus tipos de entidad subconjunto (si es total), o no (si es parcial)

Cobertura exclusiva o superpuesta

Los elementos que pertenecen a un tipo de entidad subconjunto pueden pertenecer también a otro tipo de entidad subconjunto (si es superpuesta) o no (si es exclusiva)

Ejemplo



Subclases

Existen varias buenas razones para utilizar las relaciones y subclases de ISA:

- ▶ No tiene que redefinir todos los atributos.
- ▶ Puede agregar atributos descriptivos específicos de una subclase.
- ▶ Para identificar los tipos de entidad que participan en una relación establecida de la forma más precisa posible.

Las relaciones de ISA forman una **estructura de árbol** (taxonomía) con un conjunto de entidades que actúa como **raíz**.

Principios de Diseño.

Fidelidad

- ▶ El diseño debe ser fiel a la especificación/realidad.
- ▶ Los aspectos relevantes de la realidad deben representarse en el modelo.

Evitar la redundancia

- ▶ La representación redundante explota el diagrama de ER y hace más difícil entender.
- ▶ La representación redundante desperdicia almacenamiento.
- ▶ La redundancia puede provocar inconsistencias en la base de datos.

Principios de Diseño.

Mantenlo sencillo

- ▶ Cuanto más simple, más fácil de entender para algunos lectores (externos) de los diagramas de ER.
- ▶ Evite introducir más elementos de los necesarios.
- ▶ Si es posible, prefiera atributos sobre tipo de entidades y tipo de relaciones.

Formular restricciones en la medida de lo posible

- ▶ Una gran cantidad de semántica de datos puede (y debe) capturarse.
- ▶ Pero algunas restricciones no se pueden capturar en los diagramas de ER.

Diseño de Alto Nivel con MER

Las principales opciones de diseño

- ▶ ¿Se debe modelar un concepto como una entidad o un atributo? una relación?

¿Qué relaciones utilizar: binario o ternario?

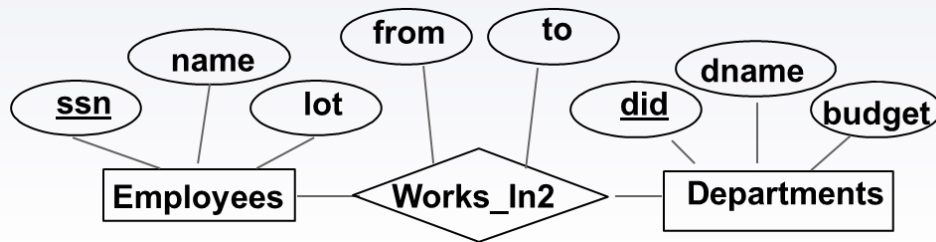
¿La dirección debe ser un atributo de Empleado o una entidad (conectada a Empleado por una relación)?

Depende del uso que queramos hacer de la información de dirección, y la semántica de los datos:

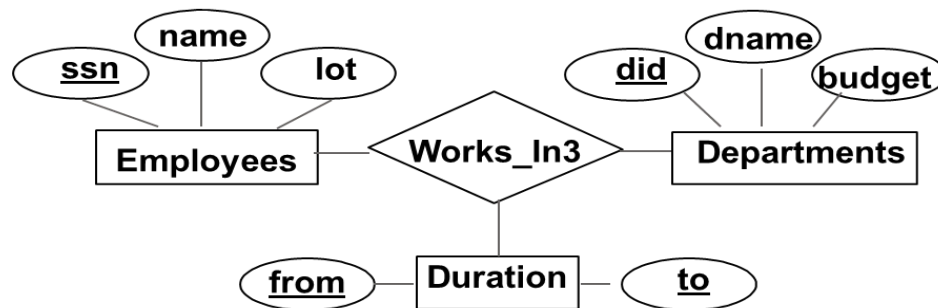
- ▶ Si tenemos varias direcciones por empleado, la dirección debe ser una entidad (ya que los atributos no pueden tener más de un valor).

Entidad vs Atributo

Works_In2 no permite que un empleado trabaje en el mismo departamento durante dos o más periodos (¿por qué?).



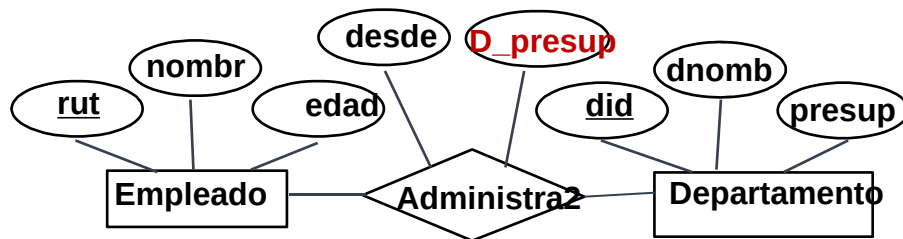
Queremos registrar varios valores de los atributos descriptivos para cada instancia de esta relación



Entidad vs Relación

Este diagrama de ER está bien si un gerente obtiene un presupuesto discrecional separado para cada departamento. Pero, ¿qué pasa si un gerente obtiene un presupuesto discrecional que cubre todos los DEPTOS administrados?

- ▶ **Redundancia de d_presup**, que se almacena para cada Departamento, gestionado por el gestor.



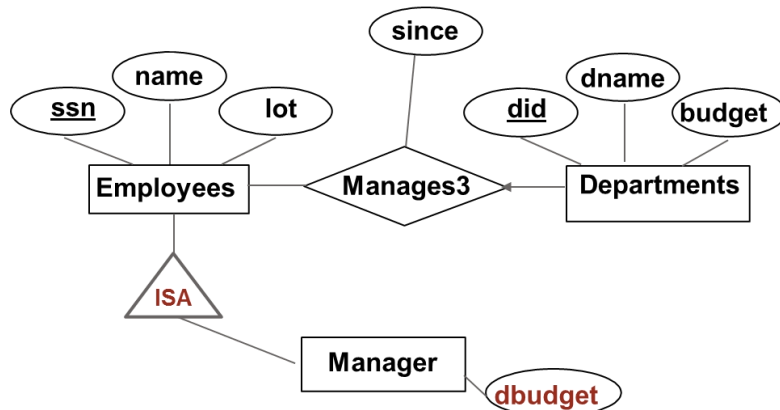
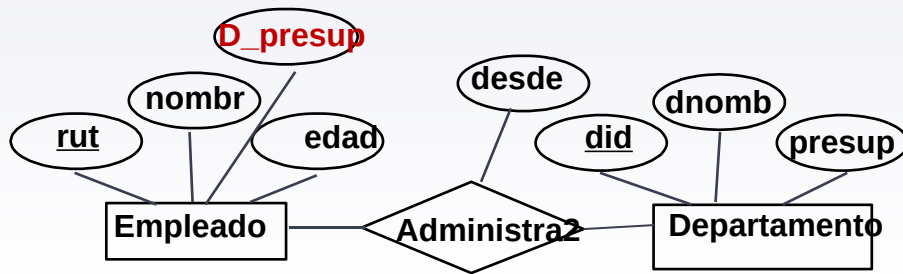
Entidad vs Relación

¿Qué hay de este diagrama?

¿Los empleados que no son gerentes tendrán dbudget = null?

El siguiente diagrama de ER es más apropiado y evita los problemas anteriores!

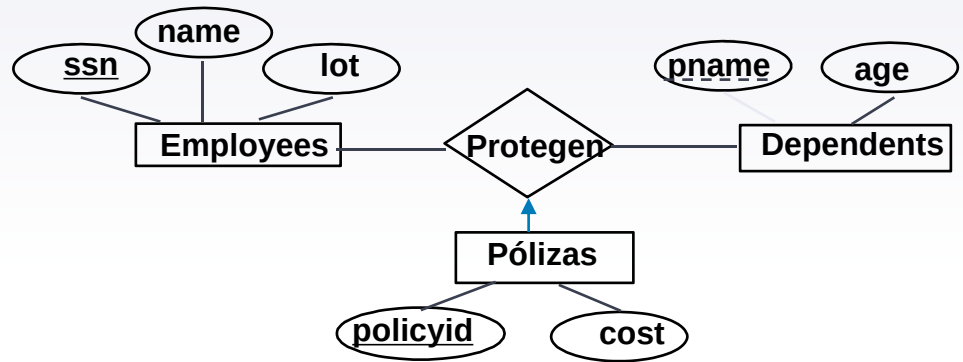
Cada Gerente tiene ahora un presupuesto.



Relaciones Binarias VS Ternarios

Con el diagrama de ER se espera:

- El empleado puede poseer varias pólizas
Cada póliza puede ser propiedad de varios empleados
- Cada dependiente puede ser cubierto por varias pólizas



Si cada póliza es propiedad de un solo empleado:

La restricción clave en las póliza significaría que la póliza sólo puede abarcar 1 dependiente! (sólo 1 combinación de *Employees* Y *Policies* puede estar en *Covers*)

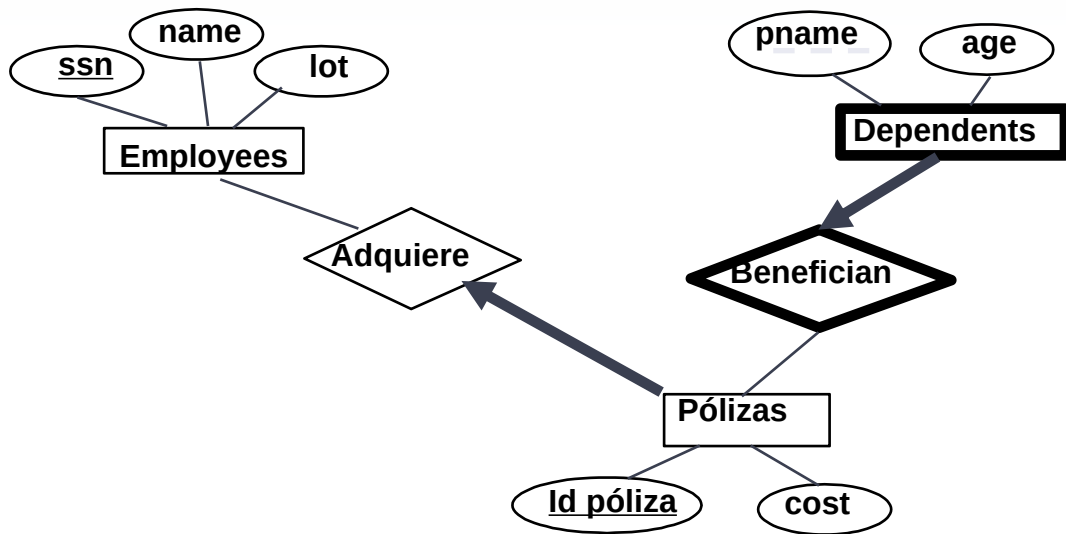
Mal diseño!

Relaciones Binarias VS Ternarias

Ternarias

Este diagrama es un mejor diseño.

- ▶ La pólizas sólo puede existir para los empleados. Los dependientes solo existen si están cubiertos por una pólizas.



Resumen

El diseño de alto nivel sigue el análisis de los requisitos y proporciona una descripción de alto nivel de los datos que se almacenarán.

MER modelo popular para el diseño de alto nivel.

El diseño de ER es subjetivo. A menudo hay muchas maneras de modelar un escenario determinado! Analizar alternativas puede ser complicado, especialmente para una gran empresa.



Universidad
de Concepción

Tecnologías de Información

Departamento de Ingeniería
Informática y Ciencias de la
Computación

